

SWINBURNE UNIVERSITY OF TECHNOLOGY

COS20007 OBJECT ORIENTED PROGRAMMING

---

## 1.1P: Preparing for Object Oriented Programming

---

PDF generated at 21:27 on Friday 21<sup>st</sup> April, 2023

# 1.1P: Preparing for OOP – Answer Sheet

1. Explain the following terminal instructions:

a. cd:

The cd command is used to change the current working directory in the terminal. It changes the directory to either the child folders or, returns the directory to the parent folder.

```
PS C:\Users\kayes>
PS C:\Users\kayes> cd C:\Users
PS C:\Users>
```

b. ls:

The ls command is used to list the files and directories in the current working directory.

```
PS C:\Users> ls

Directory: C:\Users


Mode                LastWriteTime         Length Name
----                -
d-----          5/03/2023 10:43 PM             dev
d-----          5/03/2023 10:43 PM             etc
d-----          5/03/2023 10:43 PM             home
d-----          5/03/2023 10:43 PM      installerResources
d-----         11/03/2023  2:35 PM             kayes
d-r-----        5/03/2023 10:43 PM             Public
-a-----        16/08/2022  3:46 PM          5279 InstallationLog.txt
-a-----        16/08/2022  3:46 PM           48 installer.dat
-a-----        16/08/2022  3:46 PM          362 network.xml
-a-----        16/08/2022  3:46 PM         48136 uninstall.dat
-a-----        16/08/2022  3:46 PM          5060 uninstall.ini
```

c. pwd:

It stands for print working directory. This command is used to display the current working directory in terminal.

```
PS C:\Users\kayes> pwd

Path
----
C:\Users\kayes

PS C:\Users\kayes>
```

2. Consider the following kinds of information, and suggest the most appropriate data type to store or represent each:

Information	Suggested Data Type
A person's name	String
A person's age in years	Integer
A phone number	String
A temperature in Celsius	Float
The average age of a group of people	Float
Whether a person has eaten lunch	Boolean

3. Aside from the examples already provided in question 2, come up with an example of information that could be stored as:

Data type	Suggested Information
String	A TV Series name
Integer	Number of Fingers in my hands
Float	My chest length in centimeter
Boolean	Whether I paid my rent

4. Fill out the last two columns of the following table, evaluating the value of each expression and identifying the data type the value is most likely to be:

Expression	Given	Value	Data Type
6		6	Integer
True		True	Boolean

a	a = 2.5	2.5	Float
1 + 2 * 3		7	Integer
a and False	a = True	False	Boolean
a or False	a = True	True	Boolean
a + b	a = 1 b = 2	3	Integer
2 * a	a = 3	6	Integer
a * 2 + b	a = 2.5 b = 2	7	Integer
a + 2 * b	a = 2.5 b = 2	6.5	Float
(a + b) * c	a = 1 b = 1 c = 5	10	Integer
"Fred" + " Smith"		"Fred Smith"	String
a + " Smith"	a = "Wilma"	"Wilma Smith"	String

5. Using an example, explain the difference between **declaring** and **initialising** a variable.

The difference between these two are given below:

Declaring a variable means creating a variable with a specific name and its data type in the program.

Example:

`float myHight;`

In this example, we declared a variable called "`myHight`" and its data type is Float. This is basically called declaring a variable in Programming.

On the other hand, initializing a variable means giving a value to a variable during the declaration of that variable.

Example:

`float myHeight = 1.67.`

Here in this example , we declared a variable called “myHeight” and its data type float and also initialized it with the value of 1.67.

6. Explain the term **parameter**. Write some code that demonstrates a simple use of a parameter. You should show a procedure or function that uses a parameter, and how you would call that procedure or function.

A parameter is value that passed to a method or function when it is called. Basically, works as a variable inside the method and allows the method to receive data from outside. Parameters work as variables inside the method.

Example:

```
int AddNumbers(int a, int b)
{
    int result = a + b;
    return result;
}

int sum = AddNumbers(5, 10);
```

In this example , the method is “CalAge” and it takes two int parameters ‘a’ and ‘b’. Inside of this method the two parameters are added together and stored in a new variable called “result”. To call this method we have provided two int arguments which are 5 and 10.

7. Using an example, describe the term **scope** as it is used in procedural programming (not in business or project management). Make sure you explain the different kinds of scope.

There are two main types of scope in procedural programming.

Global scope : It refers to variables and function that are accessible from anywhere in the program.

Example:

```
int global_var = 10;
```

```

void print_global_var()
{
    printf("%d", global_var);
}

int main() {
    print_global_var();
    return 0;
}

```

Local scope: Local scope, on the other hand, refers to variables and functions that are only accessible within a specific function or a loop.

Example:

```

int main() {

    int local_var = 5;
    printf("%d", local_var);
    return 0;

}

```

8. In a procedural style, in any language you like, write a function called Average, which accepts an array of integers and returns the average of those integers. Do not use any libraries for calculating the average. You must demonstrate appropriate use of parameters, returning and assigning values, and use of a loop. Note — just write the function at this point, we'll use it in the next task. You shouldn't have a complete program or even code that outputs anything yet at the end of this question.

```
def average(numbers):  
    total = 0  
    for num in numbers:  
        total += num  
    avg = total / len(numbers)  
    return avg  
  
def main():  
    average_integers = [0, 1,2,3,4,5,6,7,8]  
    avg = average(average_integers)  
    print(avg)  
  
main()
```

9. In the same language, write the code you would need to call that function and print out the result.

```
def average(numbers):  
    total = 0  
    for num in numbers:  
        total += num  
    avg = total / len(numbers)  
    return avg  
  
def main():  
    average_integers = [0, 1, 2, 3, 4, 5, 6, 7, 8]  
    avg = average(average_integers)  
    print(avg)  
  
    if avg >= 10:  
        print("Double digits")  
    else:  
        print("Single digits")
```

C:\WINDOWS\system32\cmd.exe

4.0

10. To the code from 9, add code to print the message "Double digits" if the average is above or equal to 10. Otherwise, print the message "Single digits". Provide a screenshot of your program running.



```
def average(numbers):
    total = 0
    for num in numbers:
        total += num
    avg = total / len(numbers)
    return avg

def main():
    average_integers = [0, 1,2,3,4,5,6,7,8]
    avg = average(average_integers)
    print(avg)

    if avg>=10:
        print("Double digits")
    else:
        print("Single digits")

main()
```

C:\WINDOWS\system32\cmd.exe

4.0

Single digits

Press any key to continue . . .